

Fig. 1

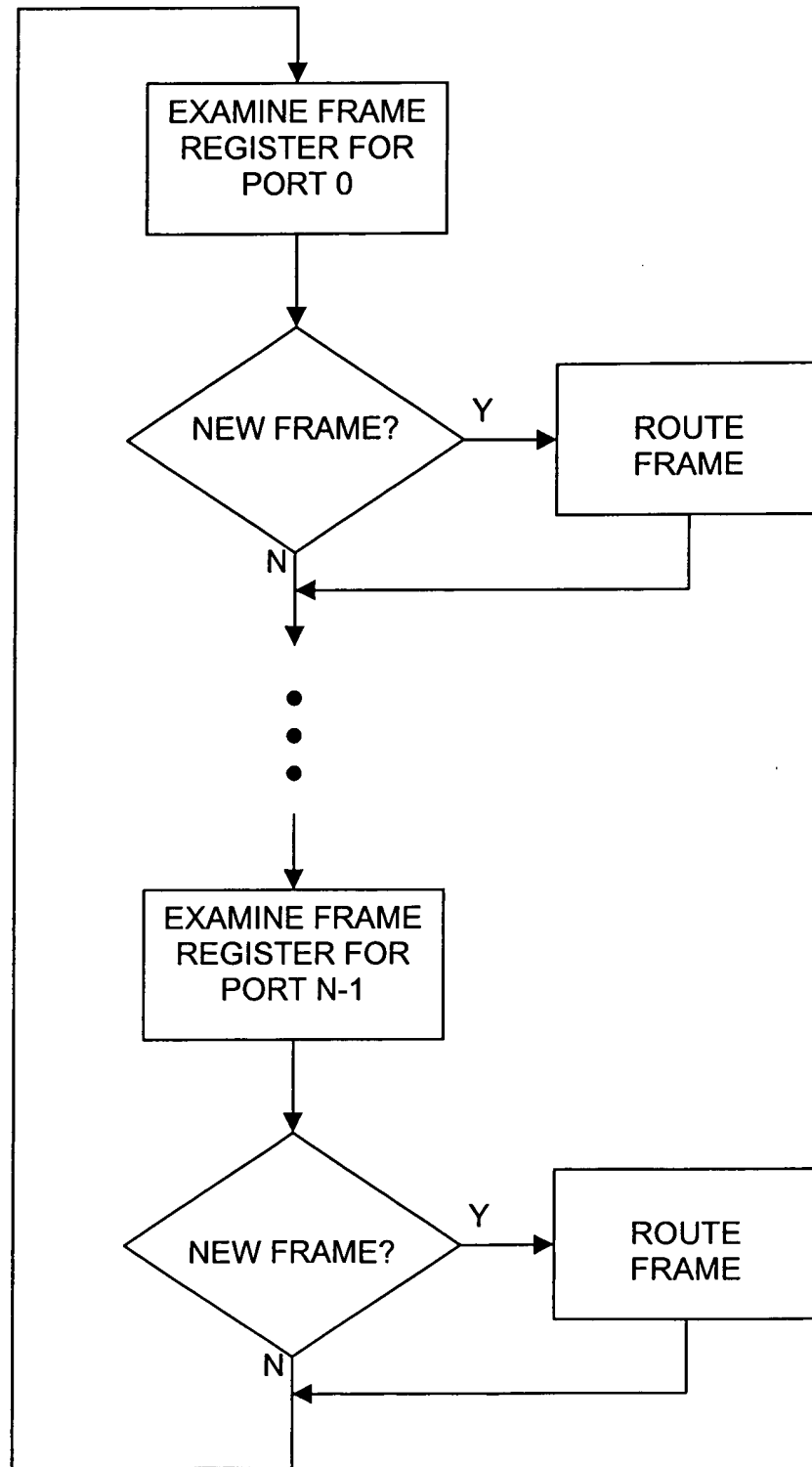


Fig. 2

```

_fast4_port0:
    ld        _Port0Hdr, r8
255    bbc.t    SOF_VLD, r8, _fast4_port1
        cmpobge.t    r8, r4, four_port0_c23
        cmpobge.t    r8, r5, four_port0_non_fastc2
four_port0_c_entry:
    st        r8, _LastFrameHeader
260    mov     0, g0
        callx    _FastBranch
        b        _fast4_port1
four_port0_non_fastc2:
    st        r8, _LastFrameHeader
265    mov     0, g0
        callx    _HandleNonFastClass2Frame
        b        _fast4_port1
four_port0_c23:
    ld        _CamResultReg, r9
270    bbc.f    BIT_FC2, r9, four_port0_c_entry
        st        r9, _Port0Dest

```

Fig. 3

```

585  /* now we copy the code for this port */
    destp = (unsigned long*)FastLoopPortPointer[port];
    srcp = (unsigned long*)(FastLoopMachineCodeBuff + (destp - (unsigned long*)FastL
for(; destp < FastLoopPortPointer[port+1]; destp++, srcp++)
{
590  *destp = *srcp;
}

/* invalidate the i960 instruction cache */
write_icctl(2);

```

Fig. 4

630

```
/* point to where the 'branch' instruction is to be written */  
destp = (unsigned long*)FastLoopPortPointer[port];
```

635

```
/* compute the offset */  
distance = ((unsigned long*)FastLoopPortPointer[port+1]) -  
            ((unsigned long*)FastLoopPortPointer[port]);
```

```
/* 0x08 is the machine code for branch. the lower 24 bit is the offset */  
*destp = 0x08000000 | (distance * 4);
```

640

```
/* invalidate the i960 instruction cache */  
write_icctl(2);  
asm("setbit 18, sf4, sf4");
```

Fig. 5

00207E" E4420260

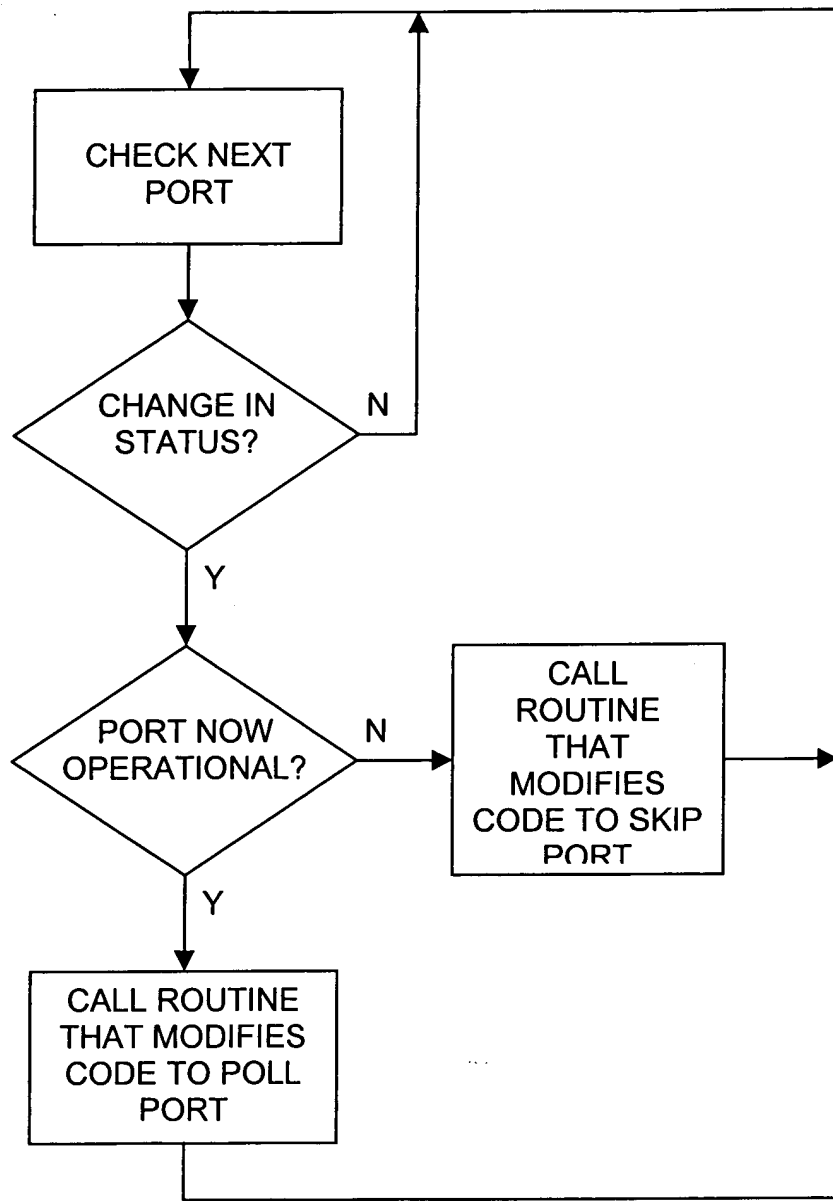


Fig. 6